

Metodología para el desarrollo de software en proyectos de I+D en el nivel universitario basada en Scrum

Luciano Straccia, Pablo Pytel, María Florencia Pollo-Cattaneo

Grupo de Estudio en Metodologías de Ingeniería de Software (GEMIS). Universidad Tecnológica Nacional. Facultad Regional Buenos Aires. Argentina.

lstraccia@frba.utn.edu.ar, ppytel@gmail.com, flo.pollo@gmail.com

Resumen. La investigación en el ámbito de las ciencias informáticas es, en muchas ocasiones, inseparable del desarrollo, constituyendo cada proyecto en un proyecto I+D (investigación y desarrollo). El desarrollo de software en el marco de un proyecto I+D debe ser una actividad llevada a cabo con estrategias y metodologías apropiadas que permitan dar respuestas a las necesidades del proyecto y que se adapten a las características institucionales y de los recursos humanos con los que se cuenta, partiendo de experiencias y herramientas brindadas por la ingeniería del software. A partir de la experiencia inicial en el desarrollo de software dentro su proyecto el grupo GEMIS, de la Universidad Tecnológica Nacional, Facultad Regional Buenos Aires (UTN-FRBA), se propuso partir de una metodología ágil y, revalorizando algunos aspectos como las definiciones funcionales y la arquitectura, definir una metodología acorde a las necesidades de proyectos de I+D en el ámbito universitario.

Palabras Claves: Desarrollo de Software, Metodología, Scrum.

1 Introducción

Dentro del ámbito de la Facultad Regional Buenos Aires de la Universidad Tecnológica Nacional (UTN-FRBA) se ha conformado el Grupo GEMIS (Grupo de Estudios de Metodología para Ingeniería de Software y Sistemas de Información), un equipo de personas con interés en la investigación vinculada a Ingeniería en Sistemas de Información y en Tecnología Aplicada a la Educación. Así se ha puesto en marcha en Mayo del 2015 el Proyecto de Investigación y Desarrollo (PID) denominado “Intervenciones tecnológicas en dispositivos didácticos con herramientas de tecnología informática” cuyo objetivo es describir y analizar el uso de la tecnología informática en las intervenciones didácticas de los profesores de las asignaturas de la carrera de Ingeniería en Sistemas de Información (ISI) de la UTN-FRBA y desarrollar nuevos artefactos tecnológicos que favorezcan la mejora en las intervenciones didácticas y una metodología de implementación.

La construcción de soluciones de Sistemas de Información y Tecnología Informática (SI/TI) para el ámbito educativo no está exento de problemas habituales

en el desarrollo de software. Sin embargo, en el marco de un PID, estos problemas tradicionales se conjugan con otros conflictos propios de la investigación en el ámbito universitario. GEMIS ha identificado algunas problemáticas específicas en su propio proyecto que requieren la definición de una metodología de desarrollo precisa y acorde a sus necesidades.

Este trabajo presenta las características del PID y los recursos humanos asociados al mismo (sección 2), realiza una descripción de la organización inicial del trabajo del equipo (sección 3), recorre las bases teóricas y metodológicas en las cuales se sostiene la propuesta de este trabajo (sección 4), presenta una metodología para el desarrollo de software en equipos de I+D (sección 5) y una evaluación de la experiencia de su implementación (sección 6). Finalmente, se indican las conclusiones obtenidas y futuras líneas de trabajo (sección 7).

2 Características del proyecto y los recursos humanos

Los proyectos de investigación y desarrollo requieren de recursos humanos que permitan continuarlos a lo largo del tiempo. La conformación de un equipo de investigadores y desarrolladores en el ámbito universitario tiene diversas fuentes de recursos posibles: docentes, alumnos y graduados. Pocos de estos recursos tienen una dedicación exclusiva o semiexclusiva al proyecto y aquellos que pudieran poseerlo se encuentran más abocados a la gestión del mismo (directores y codirectores) que al desarrollo tecnológico en sí mismo; además otros investigadores con una dedicación horaria superior a la media de los integrantes se abocan a aspectos propios de investigación que, por tratarse de trabajos asociados a la vida académica universitaria y su impacto en la didáctica, no tienen una vinculación directa con el desarrollo, excepto que son generadores de requerimientos y necesidades y acompañan la implementación de los artefactos construidos.

Indagando especialmente en cada uno de las diferentes fuentes de recursos humanos para el proyecto se ha observado además que:

- los alumnos de los primeros niveles de la carrera de ISI no poseen los conocimientos técnicos requeridos para el desarrollo de software con la complejidad que el proyecto amerita y, si lo poseen, se trata de conocimientos obtenidos por fuera del marco institucional de su carrera, como puede ser aprendizaje individual, carreras de grado previas o experiencia laboral;
- los alumnos de los niveles superiores de la carrera poseen dificultades para sostener su participación activa y continua en el proyecto por sus compromisos laborales (ya que se observa que los alumnos de niveles superiores ya poseen participación en la vida laboral en su mayoría) y esto implica, en algunos casos, el abandono de los proyectos en los que se encuentran involucrados o, en el mejor de los casos, en una disminución de su participación;
- para los graduados se observa el mismo inconveniente asociado a los alumnos avanzados.

Además, en particular el PID de GEMIS no aspira a construir un producto software único, sino diferentes aplicaciones software a través de las cuales involucrarse en la mejora de la didáctica, por lo cual no se trata de la construcción de un producto sino de múltiples productos que, integrados, tengan un impacto positivo y permitan cumplir los objetivos. En este contexto, se lleva adelante un Desarrollo de Software basado en Componentes (CBSE, *Component Based Software Engineering*), cuyo concepto se tratará más adelante.

Por otra parte, los requerimientos asociados al desarrollo son identificados o surgen a lo largo del ciclo de vida de la investigación. Los resultados obtenidos a partir de la implementación de un producto construido son fuente de análisis para el propio proyecto de investigación y permiten la definición de nuevos requerimientos. GEMIS se encuentra en un trabajo continuo para el relevamiento de dificultades asociadas al proceso de enseñanza y aprendizaje, lo que constituye un dominio de aplicación dinámico, con requerimientos cambiantes y cuyos *stakeholders* (interesados) no necesariamente conocen acerca del desarrollo de software.

Finalmente, el nivel de remuneración de los desarrolladores de los proyectos de investigación y desarrollo en la Universidad es bajo, tratándose de becas o, en un alto porcentaje, de integrantes de actividades voluntarias.

Además dado que el objetivo de GEMIS (y deseable en todo PID) involucra la formación de sus recursos humanos, se incorporan al proyecto recursos con poco *expertise* o con conocimientos tecnológicos básicos.

3 Organización inicial del equipo de desarrollo

Acorde a la necesidad de realizar en una primera etapa un relevamiento de las situaciones problemáticas existentes y, a partir de ello, generar soluciones que permitan superar estas situaciones, el inicio del proyecto se vio involucrado en una metodología de estilo cascada [1], con una definición lo más avanzada posible de requerimientos y, a partir de ello, la derivación al equipo de programación para la construcción de los respectivos productos que atendieran esos requerimientos.

El uso de esta metodología dio impulso al proyecto, permitió trabajar en los primeros aspectos de la investigación, generar definiciones acordes a lo requerido por los desarrolladores e, inclusive, a partir de clarificar el alcance vigente, la posibilidad de motivar a nuevos recursos a incorporarse al proyecto.

Sin embargo el crecimiento de las problemáticas halladas, el crecimiento de los requerimientos de software, las dificultades con los recursos ya mencionados y el retraso de las fechas de entrega, han generado la necesidad de repensar la metodología a utilizar. La metodología tradicional en cascada ha sido útil y relevante para impulsar el proyecto en sus inicios, dando a partir de su crecimiento paso a la necesidad de redefinición metodológica. A medida que el proyecto fue avanzado la definición de requerimientos formal y tradicional dio lugar al uso de historias de usuario [2] que “son descripciones cortas de una necesidad de un cliente del software que estemos desarrollando. Su utilización es común cuando se aplican marcos de trabajo ágiles” [3] y buscan favorecer la construcción del producto software sin una dedicación temporal excesiva para el análisis de los requerimientos.

El equipo de desarrollo previsto al inicio del proyecto poseía conocimientos en cierto lenguaje de programación (PHP) y los primeros productos fueron desarrollados bajo dicho lenguaje. Sin embargo, nuevos recursos humanos mostraron su interés en participar en el proyecto lo que planteaba la disyuntiva de descartar su participación o incluirlos diversificando los lenguajes de desarrollo. GEMIS ha optado por esta segunda opción ya que el lenguaje de programación (y la tecnología en general) no operan como restricciones del proyecto, excepto por su grado de factibilidad de implementación y de la disponibilidad de un servidor para su despliegue.

4 Bases teóricas y metodológicas de la propuesta

En este capítulo se presentan las bases teóricas y metodológicas en las cuales se sostiene la propuesta del presente trabajo. En una primera parte se describen las características del Desarrollo de software basado en Componentes (sección 4.1), luego se describen generalidades del desarrollo ágil (sección 4.2), la metodología Scrum (sección 4.3) y el papel de la arquitectura de software (sección 4.4). Finalmente se plantea la Gestión de personas como pool de recursos (sección 4.5).

4.1. Desarrollo de Software basado en Componentes

El desarrollo de software basado en componentes (CBSE, *Component Based Software Engineering*) es aquel que está fundamentado en la producción de diversas piezas de software ensambladas de una manera integral que permita el funcionamiento del sistema software como un todo. En este contexto, se define a un sistema como “un conjunto de mecanismos y herramientas que permiten la creación e interconexión de componentes software, junto con una colección de servicios para facilitar las labores de los componentes que residen y se ejecutan en él” [4].

Un componente es una pieza de código preelaborado que encapsula alguna funcionalidad expuesta a través de interfaces estándar. Szyperski, exarquitecto de software de Microsoft, define componente como una “unidad de composición de aplicaciones software que posee un conjunto de requisitos y que ha de poder ser desarrollado, adquirido, incorporado al sistema y compuesto por otros componentes, de forma independiente en tiempo y forma” [5]. Entre las ventajas que pueden hallarse en el uso del CBSE se encuentra la reutilización de software, la simplificación de las pruebas, la simplificación del mantenimiento del sistema y una mayor calidad de los componentes [6].

El aspecto de mejora de la calidad es una variable fundamental que ha llevado a GEMIS a implementar esta estrategia de desarrollo de software, dado que permite la construcción de un componente que luego puede ser reemplazado (o mejorado) por un nuevo componente, el cual podría ser desarrollado por un programador con más *expertise* que el programador original.

4.2. Desarrollo ágil

Poole [7] define el desarrollo ágil “como aquel que, en comparación con el desarrollo tradicional, provee beneficios de mayor flexibilidad, retorno de inversión más alto, realización más rápida del retorno de la inversión, más alta calidad, (y) mayor visibilidad” [8]. El desarrollo ágil no sólo ha otorgado diferentes metodologías (Scrum, XP, ASD, Crystal, entre otras) sino que también ha brindado técnicas y herramientas aplicables independientes de la metodología (algunas de las cuales han sido inicialmente aplicadas en el equipo GEMIS).

El Manifiesto Ágil [9] reúne las motivaciones y principios de un desarrollo ágil, entre los cuales se encuentra la satisfacción del cliente con entrega temprana y continua de software, considerando al software funcionando como la medida principal de progreso; adaptación a requisitos cambiantes; confianza en los individuos (con el entorno y apoyo necesario) y simplicidad en el desarrollo. Por otro lado, a partir de las características de GEMIS cobran especial relevancia algunos otros aspectos expresados en el Manifiesto: valoración de la comunicación (con conversaciones cara a cara); promoción del desarrollo sostenible por medio de los procesos ágiles; mejora en las arquitecturas, surgimiento de requisitos y soluciones a partir de la actividad de equipos auto-organizados; y reflexión periódica del equipo sobre cómo ser más efectivo para, a continuación, ajustar y perfeccionar su comportamiento en consecuencia.

4.3. Scrum

Scrum es una metodología ágil que se basa “en la teoría de control de procesos empírica o empirismo. El empirismo asegura que el conocimiento procede de la experiencia y de la toma de decisiones basándose en lo que se conoce. Scrum emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control del riesgo” [10].

El equipo Scrum se compone de un responsable del producto (dueño del producto, *Product Owner*), el equipo de desarrollo (*Development Team* o *Scrum Team*) y el *Scrum Master*, que es responsable de asegurar que Scrum es entendido y adoptado correctamente.

Scrum supone los siguientes artefactos: lista del producto (*Product Backlog*), lista del sprint (*Sprint Backlog*) y un incremento. Un *sprint* es cada iteración del proceso de desarrollo. La lista del producto contiene todas las funcionalidades previstas para el producto y la lista del *sprint* contiene el listado de aquellas funcionalidades que serán incluidas en la iteración siguiente (sin considerar ni definir en cuál sprint serán incluidas las restantes). El incremento es “la parte de producto producida en un sprint, y tiene como característica el estar completamente terminada y operativa, en condiciones de ser entregada” [11], independientemente de si finalmente el *Product Owner* decida realizar la entrega al cliente o destinatario del mismo.

Los *sprints* comienzan con una reunión donde se planifica el mismo (*Sprint Planning Meeting*) y finalizan con una revisión del sprint en cuanto al producto (*Sprint Review*) y una retrospectiva (*Sprint Retrospectiva*), que implica una revisión del sprint en cuanto a personas, relaciones, procesos y herramientas [10]. Durante el sprint se llevan a cabo reuniones diarias (*Daily Meeting*) donde se comparte información sobre el estado de la iteración y las dificultades presentes.

4.4. El papel de la arquitectura de software

La arquitectura del software es la organización fundamental de un software, formada por sus componentes, las relaciones entre ellos, el contexto en el que se implantarán, y los principios que orientan su diseño y evolución [12]. La bibliografía asociada a Scrum generalmente no hace referencia a las decisiones arquitecturales y las discusiones existentes acerca de cómo incorporar la arquitectura en los proyectos dirigidos por Scrum son confusos. Dado que muchos adeptos a esta metodología provienen del campo de la programación del software, se pueden hallar manifestaciones acerca de la arquitectura como emergente del código fuente. Sin embargo, si bien la arquitectura puede emerger del código (uno los principios del manifiesto ágil se vincula a los diseños emergentes de los equipos de desarrollo), es importante la formalización (que no implica necesariamente documentación ni aspectos generalmente definidos como “burocráticos”) de la arquitectura a fines de poder ser compartida por la totalidad del equipo, especialmente cuando se espera llevar a cabo de un desarrollo basado en componentes.

Mago y Alferez [13] proponen incorporar la arquitectura del software en el proceso de Scrum como un *sprint* 0, en el cual se define la arquitectura que guiará el proyecto; sin embargo esto no es factible utilizarlo cuando se piensa el software con una arquitectura cambiante ya que los autores prevén que el arquitecto de software guíe al equipo en los *sprints* subsiguientes a fines de orientarlos para la adaptación a la arquitectura predefinida, pero no la readaptación o reestructuración arquitectónica.

4.5. Gestión de personas como *pool* de recursos

La dirección de las actividades de desarrollo del proyecto implica gestionar una cartera de proyectos constituyendo una responsabilidad similar a un PMO (*Project Management Office*, Oficina de Gestión de Proyectos) que es definida por el PMBOK® como “una unidad de la organización para centralizar y coordinar la dirección de proyectos a su cargo” [14], siendo una unidad diseñada para dirigir y controlar el desarrollo de un grupo de proyectos informáticos de manera simultánea. Considerando la caracterización de PMO realizada por el Project Management Institute (PMI), GEMIS trabaja con el marco de trabajo correspondiente a una oficina de respaldo, servicios y controles del proyecto que “proporciona los procesos que faciliten un apoyo continuo para las labores de gestión del proyecto, programa o cartera en toda la organización. Aplica la gobernanza, procesos, prácticas y herramientas establecidas por la organización y brinda un apoyo administrativo para las labores del proyecto, programa o cartera dentro de su dominio” [15].

Casey y Peck [16] plantean tres modelos de PMO: tipo estación meteorológica, torre de control y pool de recursos, en las cuales el foco principal de trabajo son la generación de informes de acompañamiento de indicadores, el control de proyectos y gestión del conocimiento y la gerencia y aplicación de recursos, respectivamente [17]. GEMIS adopta, por las características de sus proyectos y sus recursos, una dirección asociada al pool de recursos, también denominada en ocasiones Escuadrón de Combate [17]. Debe tenerse en cuenta que la metodología implantada por GEMIS no considera gerentes de proyectos, por lo cual la asignación por parte de la PMO de recursos a los proyectos implica la asignación directa bajo el modelo de pool de

recursos como un inventario de recursos disponibles. Según el DRAE, un pool es un “grupo de personas entre las que se reparte una tarea determinada” [18].

5 Metodología propuesta

GEMIS ha considerado importante, a partir de las experiencias obtenidas en la organización inicial del equipo de desarrollo (descritas en la sección 3) y las características del proyecto y los recursos humanos existentes (descritos en la sección 2), la definición de una metodología basada en Scrum y adaptada a los requerimientos propios del equipo: desarrollar software basado en componentes, *sprints* que superen los tiempos previstos en las reglas del Scrum (donde generalmente se considera mandatorio una ventana de tiempo máxima de 1 mes) y revalorizar el papel de cierta documentación funcional como mandatoria y una definición cambiante de arquitectura de software.

El aspecto funcional cobra mayor valor por la necesidad de comunicar detalles respecto del dominio a la comunidad académica como parte de las responsabilidades de un grupo de investigación y desarrollo de una universidad. En tanto el valor de los aspectos arquitectónicos ha sido desarrollado previamente al hacer referencia al desarrollo basado en componentes.

Ambos aspectos se han incorporado a los requerimientos del incremento, así el resultado del sprint consiste en un incremento de software, un documento de arquitectura y una breve descripción de la funcionalidad incorporada, realizada bajo la forma de historias de usuario, incorporándose estos productos a los artefactos resultantes (Figura 1)

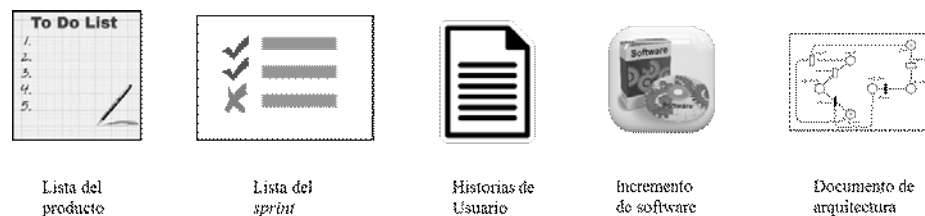


Figura 1: Artefactos GEMIS

La lista de producto y la lista de sprint, junto con la información general del producto, conforman para GEMIS un documento denominado Visión de Producto y del cual puede hallarse una plantilla en <https://goo.gl/HgHBuP>.

El proceso de Scrum no se ve modificado respecto al proceso original, exceptuando las duraciones de los *sprints* llevando cada sprint a una duración de 3 meses, período definido para la *Scrum Meeting* y la periodicidad de las *Daily Meeting* que dejan de ser diarias y pasan a ser semanales (*Weekly Meeting*).

Los *sprints* de cada uno de los productos se llevan a cabo en fechas idénticas, de manera de permitir utilizar el *Scrum Meeting* para la reasignación de recursos provenientes del pool de recursos con el que se cuenta, posibilitando incorporar a una

nueva iteración de un producto un recurso que estuviera, hasta la iteración anterior, involucrado en otro proyecto.

La asignación de recursos a cada uno de los proyectos se hace a partir de la presentación del estado general de cada uno de los productos y la revisión de productos por crearse o tareas asociadas a investigación tecnológicas (listado proporcionado en una Visión de Proyecto y que es discutido en un *Project Meeting*, coincidente en tiempo y espacio con el *Scrum Meeting* (Figura 2)). De esta manera, en el *Scrum Meeting* se presentan los diferentes productos existentes y su estado de avance, los requerimientos pendientes y los recursos podrá seleccionar en cuál producto desean continuar su participación siendo asignados al mismo previo a la generación de un nuevo *sprint*, constituyendo un *Project Meeting*.

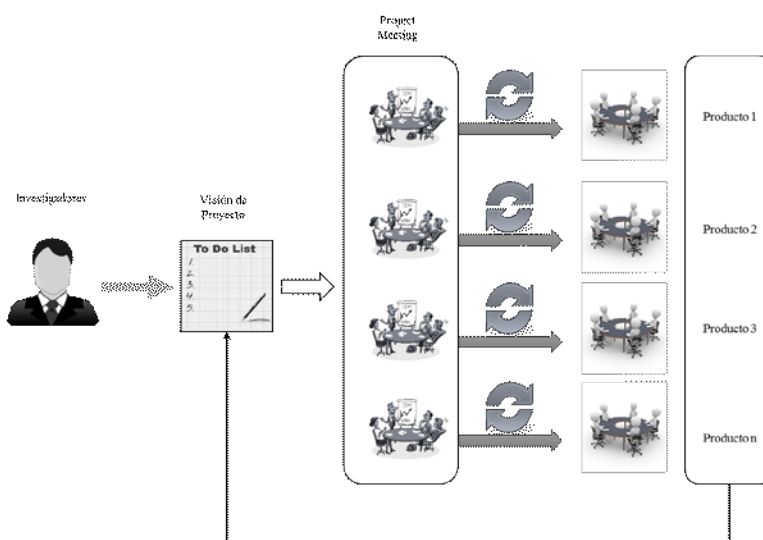


Figura 2: Metodología GEMIS

6 Experiencia de implementación

Desde el inicio del PID se han presentado diversos trabajos que avalan la perspectiva de trabajo del grupo GEMIS [19],[20],[21] y se inició el desarrollo de diversos productos: un sitio web para el análisis de evaluaciones en el ámbito universitario [22], desarrollo de soluciones móviles dirigidas a la evaluación diagnóstica como parte del proceso de enseñanza y a la resolución de problemas [23],[24] y el desarrollo de una aplicación móvil para la autoevaluación por parte del alumno [25].

Si bien los desarrollos previstos han mostrado avances, se tornaba dificultoso constituir versiones estables de las aplicaciones y los tiempos reales siempre superaban, en gran medida, a los tiempos previstos. A partir de la nueva perspectiva de trabajo y la implementación de la propuesta metodológica explicada en este trabajo

se ha logrado configurar un mapa de aplicaciones en las cuales, incluso, a partir de las discusiones generadas en las primeras reuniones de equipo bajo esta nueva metodología, se logró acordar un nuevo componente, denominado Datos Maestros (DM), que prestaría servicios a las diferentes aplicaciones, definiéndose que se realizaría como Servicios REST.

Respecto de cada producto, se ha logrado llegar a una documentación precisa respecto de los alcances esperados, pudiendo constituir todos los artefactos definidos en la metodología y posibilitando la disponibilidad de software funcionando que se encuentra en las primeras etapas de puesta en producción. El producto CW que se inició en el año 2014 tuvo gran deficiencia en la construcción del software ya que, a pesar de presentarse algunos avances, estos no cumplían con las expectativas. A partir de la puesta en marcha de la metodología, se logró finalizar una primera versión del sitio web (primer *sprint*) e incorporar el módulo de reportes (segundo *sprint*) encontrándose actualmente en el desarrollo de su tercer *sprint*. Asimismo, se logró poner en marcha un proyecto de centralización de datos maestros (DM) y generar una primera versión de los productos RR y TD [23],[24], constituir el equipo necesario para el inicio de un nuevo producto que se encuentra transitando su primer *sprint* (producto AS, Asistencia) e integrar el trabajo de GEMIS con el trabajo de una institución terciaria con carreras asociadas al desarrollo de software tal como el Instituto Superior Da Vinci a través del cual se pudo iniciar el producto IA (Intercambio de Archivos), que se encuentra también en el primer *sprint*.

Finalmente, la metodología propuesta en el presente trabajo permitió la incorporación de nuevos recursos humanos al proyecto que en principio no tenían una asignación prevista pero su participación en las *Project Meeting* permitió asignarlos a algunos de los productos con desarrollo vigente.

7 Conclusiones

Los proyectos de I+D en el ámbito universitario asociados al desarrollo de software no son ajenos a las problemáticas habituales existentes en la industria del software. Asimismo incorporan problemáticas propias del quehacer institucional universitario, las particularidades de los recursos humanos y nuevas obligaciones respecto a resultados esperados.

En este trabajo se presenta la adaptación de la metodología Scrum para el desarrollo de software para proyectos de I+D en el ámbito universitario, considerando especialmente la importancia del Desarrollo de Software Basado en Componentes, el papel de la arquitectura de software y la gestión de personas como pool de recursos.

A partir de la puesta en marcha de la metodología se produjeron avances importantes sobre uno de los productos iniciados (CW), se logró poner en marcha un proyecto de centralización de datos maestros (DM), generar una primera versión de nuevos productos (RR y TD), iniciar un nuevo producto (AS) y trabajar integradamente con otras instituciones

Como futuras líneas de trabajo se hace necesario continuar con la aplicación metodológica en los diferentes proyectos y productos existentes y verificar el cumplimiento de las diferentes metas planteadas. Además dadas las características de

los recursos humanos, será necesario realizar una gestión de los recursos no sólo para resolver problemas y necesidades presentes sino también previendo potenciales cambios y necesidades futuras, bajo una perspectiva de gestión del pool de talentos [26].

Referencias

1. Kimball, R., et al. The Data Warehouse Lifecycle Toolkit. John Wiley & Sons. (2011)
2. Ambler, S. User Stories: An Agile Introduction. (2014)
3. PMOInformatica. Historias de usuario en 5 pasos. (2014)
4. Fuentes, L.; Troya, J.M.; Vallecillo, A. Desarrollo de Software Basado en Componentes. ETSI Informática. Universidad de Málaga. Málaga, España.
5. Szyperski, C. Component Software. Beyond Object-Oriented Programming. Addison-Wesley. (1998)
6. Casal Terreros, J. MSDN Microsoft. Desarrollo de Software basado en Componentes.
7. Poole, D. Do It Yourself Agile. (2009)
8. Gimson, L. Metodologías ágiles y desarrollo basado en conocimiento. Universidad Nacional de La Plata. (2012)
9. Manifiesto por el Desarrollo Ágil de Software. Disponible en <http://agilemanifesto.org/iso/es/>
10. Schwaber, K. y Sutherland, J. La Guía Definitiva de Scrum: Las reglas del Juego. (2013)
11. Scrum Manager. Disponible en www.scrummanager.net
12. IEEE Std. 1471-2000.
13. Mago, E.; Harvey Alferez, G. El Papel de la Arquitectura de Software en Scrum. Revista Software Gurú Nro. 30. Noviembre 2010 - Enero 2011. Venezuela. (2011)
14. PMBOK. Guía de los Fundamentos de la Dirección de Proyectos (Guía del PMBOK®). Tercera Edición. Project Management Institute, EE.UU. (2004)
15. PMI. Informe Pulso de la Profesión: marcos de trabajo de la PMO. PMI. (2013)
16. Casey, W. y Peck W. Choosing the right PMO setup. En Network, Febrero (2001).
17. Alonso González, A. Como implantar una oficina de gestión de proyectos en su organización. Visión Libros. Madrid. (2008)
18. DRAE. Diccionario la Real Academia Española.
19. Straccia, L.; Vegega, C.; Bernal, L.; Deroche, A.; Pytel, P.; Pollo-Cattáneo, M.F. Intervención tecnológica no convencional en los dispositivos didácticos de la carrera de Ingeniería en Sistemas de Información. XVII WICC. Universidad Nacional de Salta. (2015)
20. Straccia, L.; Acosta, M.; Pytel, P.; Pollo-Cattáneo, M.F. Intervenciones tecnológicas no convencionales en dispositivos pedagógicos. En Libro de Trabajos IV Jornada de Enseñanza de la Ingeniería. UTN, Avellaneda. (2014)
21. Straccia, L.; Vegega, C.; Pytel, P.; Pollo-Cattáneo, M.F. Tecnología Informática para facilitar la labor docente en la formación por competencias en Ingeniería. XII Congreso Internacional sobre el Enfoque basado en Competencias. Cartagena de Indias. (2016)
22. Straccia, L.; Deroche, A.; Vegega, C.; Pytel, P.; Pollo-Cattáneo, M.F. Software para análisis de instrumentos de evaluación en la formación por competencias. Proceedings XX CACIC. Universidad Nacional de La Matanza. (2014)
23. Straccia, L.; Acosta, M.; Vegega, C.; Pytel, P.; Pollo-Cattáneo, M.F. Mejora de la didáctica en la enseñanza de la Ingeniería a través del uso de nuevas aplicaciones sobre los dispositivos móviles. TEYET. Univ Nacional del Noroeste. (2015)

24. Straccia, L.; Marino Aguirre, M.; Acosta, M.; Vegega, C.; Pytel, P.; Pollo-Cattáneo, M.F. El desarrollo de artefactos de tecnología informática como aporte a las intervenciones didácticas. III CONAIISI. UTN. Buenos Aires. (2015)
25. Deroche, A.; Acosta, M.; Vegega, C.; Bernal Tomadoni, L.; Straccia, L.; Pytel, P.; Pollo-Cattáneo, M.F. Diseño de aplicación móvil para asignatura de grado en Ingeniería en Sistemas de Información. III CONAIISI. UTN. Buenos Aires. (2015)
26. MalikehBeheshtifar; Fateme-BegomKamani-Fard. Talent Pool: A Main Factor to Success. En Interdisciplinary Journal of Contemporary Research in Business. Vol. 4. Nro 12. Abril 2013.